PEGASIS: Practical Effective Class Group Actions

CRYPTO 2025 - Santa Barbara

R. Invernizzi - joint work with Dartois, Eriksen, Fouotsa, Le Merdy, Robert, Rueger, Vercauteren and Wesolowski

August 18

Group Actions

Group actions provide *quantum secure* "replacement" for DLP:

Definition (Group Action)

Given a group G and a set X, a *group action* $G \curvearrowright X$ is a map

$$\begin{aligned} G \times X &\to X \\ (g, x) &\mapsto g \star x \end{aligned}$$

compatible with group operations in G.

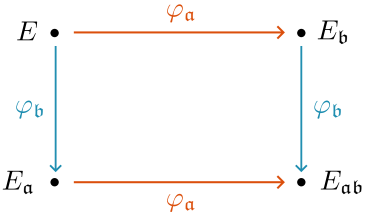
Group Actions

- if $G = \mathbb{Z}$ and $X = \mathbb{F}_p^*$ then $G \curvearrowright X$ by $(g, x) \mapsto x^g$
- *efficient evaluation*: given (x, g) can compute $g \star x$
  - classically: *exponentiation*
- *vectorization*: given $(x, g \star x)$ it is hard to recover g
  - classically: *DLP*

fwo KU LEUVEN

Isogeny Based Group Actions: CSIDH

- action of ideals of $\mathbb{Z}[\sqrt{-p}]$ on *supersingular elliptic curves*
- action given by *isogenies*: $\mathfrak{a} \star E = \varphi_{\mathfrak{a}}(E)$
- equivalent ideals result in the same curve



REGA vs EGA

- CSIDH can act efficiently only with *smooth norm ideals*
- gives a Restricted Effective Group Action (REGA) instead of EGA
- leads to problems when uniform sampling is required (e.g. *signatures*)
- CSI-FiSh / Scallop: EGA for small parameters with precomputations
- Clapoti: polynomial time framework, no practical instantiation
- KLaPoTi: Clapoti for small parameters, does not apply to CSIDH

Setup

- working over base field $\mathbb{F}_p$
- need $p = f2^e - 1$ with f small for efficiency reasons (*HD isogenies*)
- subexponential quantum attack by *Kuperberg*
- requires p to be $500 - 4000$ bits (debated)

The Clapoti Framework

Unrestricted group actions in *polynomial time*:

- take any ideal $\mathfrak{a} = (l, \sigma)$
- find two *equivalent ideals* $\mathfrak{b}, \mathfrak{c} \sim \mathfrak{a}$ and integers u, v such that

$$uN(\mathfrak{b}) + vN(\mathfrak{c}) = 2^e$$

  with $2^e < p$
- compute a degree u and a degree v isogeny
- magically (but *efficiently*) recover $\varphi_{\mathfrak{a}}$

Problem 1: solvability of equation

- expected # solutions: $2^e/N(\mathfrak{b})N(\mathfrak{c})$
- need $N(\mathfrak{b}), N(\mathfrak{c})$ as small as possible
- $\mathfrak{a}$ is a lattice, equivalent ideals given by short elements in $\mathfrak{a}$
- by *Minkowski*, smallest equivalent ideals have norm $\approx \sqrt{p}$
- expected solutions $\approx 1/4f$, *already low*

Problem 2: isogenies of given degree

- degree u, v isogenies from a random curve E are *conjecturally hard*
- if $u = x^2 + y^2$ we have a u-isogeny

$$\Phi_u = \begin{pmatrix} x & y \\ -y & x \end{pmatrix} : E \times E \to E \times E$$

- Φ_u is a dimension 2 isogeny, so Clapoti runs in *dimension 4*

Problem 2: sums of squares

- need u, v to be *sums of squares* to compute isogenies
- $u = \prod p_i^{e_i}$ is a sum of squares iff for $p_i \equiv 3 \bmod 4$, e_i is even
- *full factorization* is too expensive
- trial division and *hope leftover part is prime* instead
- $u, v \approx \sqrt{p} \rightarrow$ the probability of two sums of squares is $\approx 1/\log(p)^2$

Clapoti in practice

- forced to take $2^e < p$ for efficiency
- equation barely has a solution
- e.g. 97% *failure rate* for 4000 bit prime
- u, v being sums of squares reduces probability of $2-3$ orders of magnitude
- success probability *close to* 0

Our approach

- taking out small degree isogenies solves *both* problems
- can be computed using *Elkies algorithm*
- can evaluate $\varphi_{\mathfrak{a}}(E)$ for $\mathfrak{a} = (l, \sigma)$
- complexity $O(N(\mathfrak{a})) = O(l)$ so only for *small degree* isogenies

Problem 1: solvability of equation

- write $\mathfrak{b} = \mathfrak{b}_e \mathfrak{b}_k$ and $\mathfrak{c} = \mathfrak{c}_e \mathfrak{c}_k$ with $N(\mathfrak{b}_e)$ and $N(\mathfrak{c}_e)$ *smooth*
- apply $\mathfrak{b}_e, \mathfrak{c}_e$ using Elkies algorithm
- solve $uN(\mathfrak{b}_k) + vN(\mathfrak{c}_k) = 2^e$
- $\#$ solutions $\approx N(\mathfrak{b}_e)N(\mathfrak{c}_e)/4f$, *increased by* $N(\mathfrak{b}_e)N(\mathfrak{c}_e)$

Problem 2: sums of squares

- for small l, $\mathfrak{a}_l = (l, \sigma)$ and $\varphi_l = \varphi_{\mathfrak{a}_l}$ has degree l
- can take out from u, v *small factors* that are $3 \bmod 4$
- heuristically, removing $3, 7, 11$ already *increases success rate* $\times 3$

Rerandomization

- some ideals may still fail or take too long
- we can *rerandomize* bad ideals
- solve the norm equation for $\mathfrak{a}\mathfrak{a}_3$ instead of $\mathfrak{a}$
- then apply φ_3 to go back
- *zero failure* and *more efficient* in practice

Timings

		Size p (bits)				
Paper	Language	500	1000	1500	2000	4000
SCALLOP	C++	35 s	750 s	-	-	-
SCALLOP-HD (2D)	Sage	88 s	1140 s	-	-	-
PEARL-SCALLOP	C++	30 s	58 s	710 s	-	-
KLaPoTi (2D)	Sage	207 s	-	-	-	-
	Rust	1.95 s	-	-	-	-
PEGASIS (4D)	Sage	1.53 s	4.21 s	10.5 s	21.3 s	121 s

PEGASIS timings breakdown

- Step 1: solve norm equation
- Step 2: small degree isogenies
- Step 3: 4D isogeny

Size p (bits)	Step 1	Step 2	Step 3	Total	Rerand.
500	0.097 s	0.48 s	0.96 s	1.53 s	0.17
1000	0.21 s	1.16 s	2.84 s	4.21 s	0.07
1500	1.19 s	2.85 s	6.49 s	10.5 s	1.53
2000	1.68 s	8.34 s	11.3 s	21.3 s	0.70
4000	15.6 s	52.8 s	53.5 s	122 s	0.41

Open questions

Q1: can we solve $N(\mathfrak{b}) + N(\mathfrak{c}) = 2^e$ (no need for u, v)?

- *yes* (coming soon *qt-Pegasis*)

Q2: can we solve $N(\mathfrak{b}) + N(\mathfrak{c}) = 2^b$ for $b < e$?

- expect a solution for $2e/3$ but don't know how to find it

Q3: do we really need dimension 4?

- new ideas needed!

Thank you for your attention.